Software Requirements Specification

for

Face Recognition System [Database Module]

Version 1.0

Prepared by
Empire team

Jilin University

10 Apr 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Limeng Qiao	2016/04/10	Create	0.1
Limeng Qiao	2016/04/13	Improve	0.2

1. Introduction

1.1 Purpose

This SRS describes the module function and nonfunctional requirements for version 1.0 of the Database Module. This document is intended to be used by team members whom will implement and verify the system. In addition, this SRS is aimed at facilitating the communication with other teams of Face Recognition Project.

1.2 Intended Audience and Reading Suggestions

The readers that the documents is intended for include architect, developers, project managers, users, and documentation writers. The rest of this SRS is organized in the following way. In the next section.

1.3 Product Scope

Database Module is a module in Face recognition project, to create software for storing user's information (i.e. username, MD5 value of password, recognition websites info and so on). Database module can receive requests from Authentication server module and return validation information or error code by JSON code to Authentication server module.

1.4 References

1. SRS template from (http://www.csc.villanova.edu/~tway/courses/csc4181/s2010/srs_template-1.doc)

2. Overall Description

2.1 Module Perspective

Database module is a sub system of Face Recognition System, which will be to develop and extend an application framework using facial recognition to support remote authentication. This module is mainly composed of three parts including TEST, SIGN IN, and SIGN UP and uses protocols with JSON format to interact with authentication server. A simple diagram that shows the major components of the sub system by Figure 1.

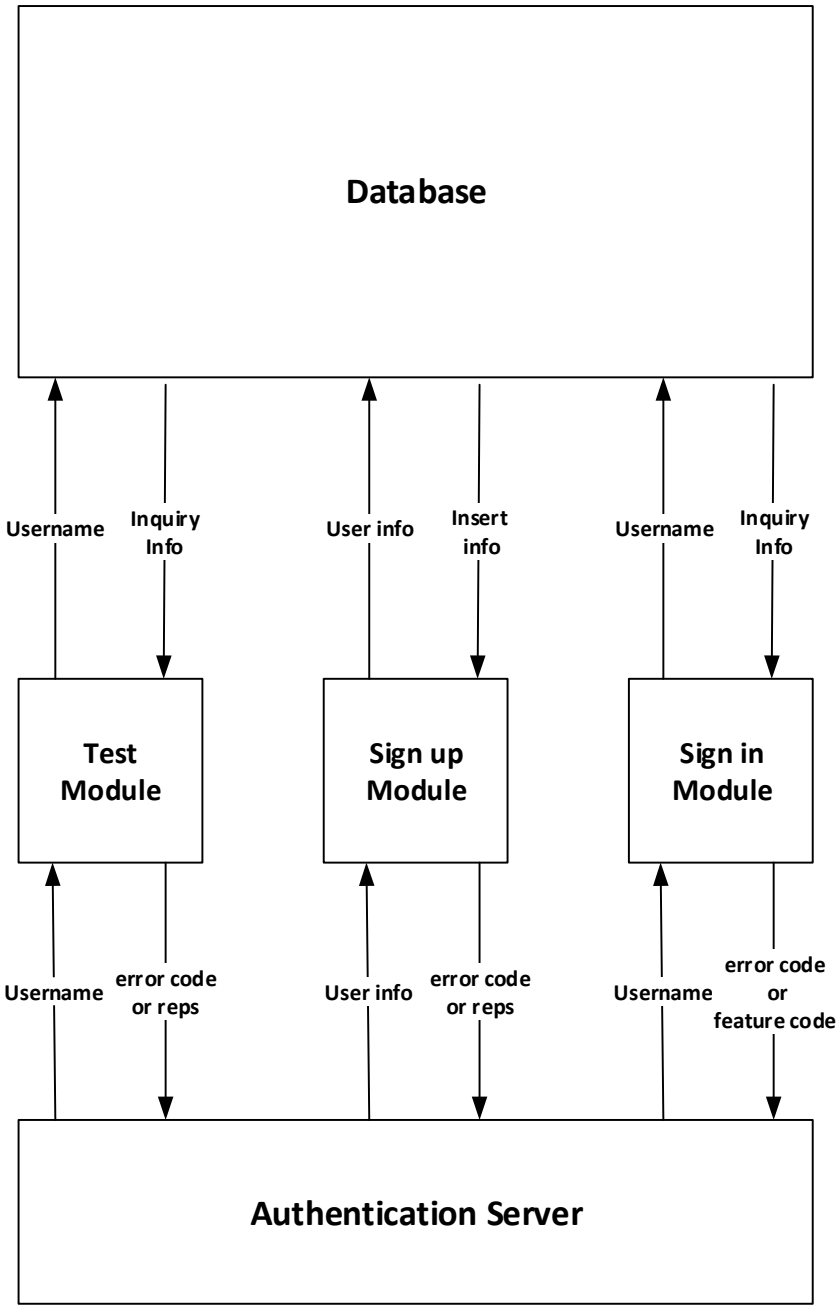

Fig. 1. The diagram of interaction between Database and Authentication Server

Module Functions

The TEST module is used to test whether current username exists in the database or not. If the username is the first time appeared in the database, TEST will return code 200 (i.e. current username is valid) or code 201 (i.e. which tells server that the current username is existent).

The SIGN UP module receives the user information from server and creates a record in the database for each user. This module is executed after module TEST, which guarantees that current info is legal. What's more, the module will performs an insert operation on the database to store user information. The SIGN UP module return code 200 (i.e. the insert operation is successful) or code 202 (i.e. which means that an unknown error occurred from database).

The SIGN IN module receives the username from server and return code to tell server whether the user is successful. What's more, the module will performs a search operation on the database to check the user's information. If OK, the database module will return feature code as current user's identification code, user's password (i.e. using MD5 encryption), and code 200 to server. Otherwise, the database module return code 203(i.e. user not registered error) or code 404 (i.e. other unknown errors).

2.2 User Case Diagram

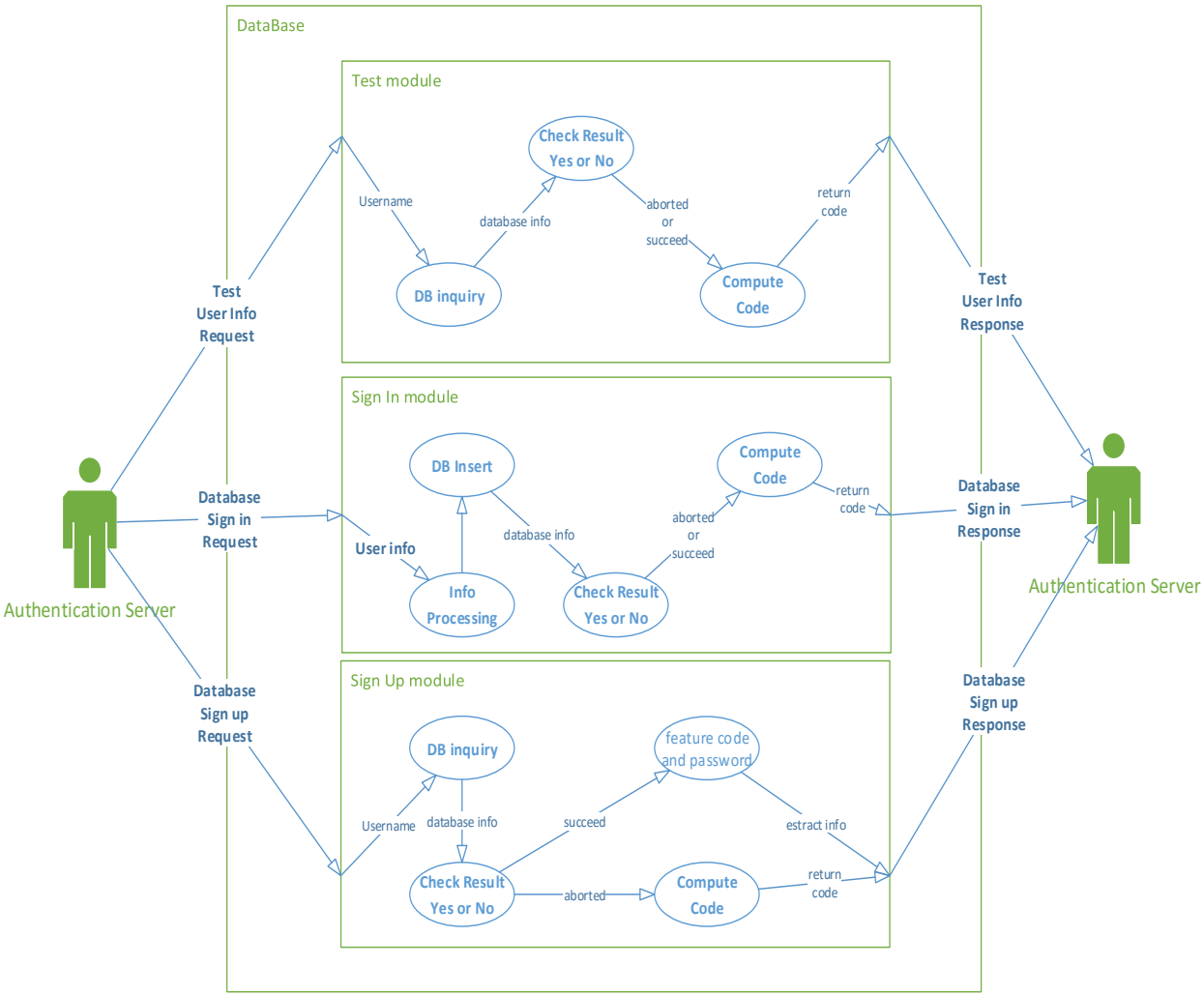

Fig.2. User case Diagram for Database

2.3 Data Flow Diagram

According to the User Case Diagram, the database module need to maintain the following data flow diagram (see figure 3, figure 4 and figure 5).

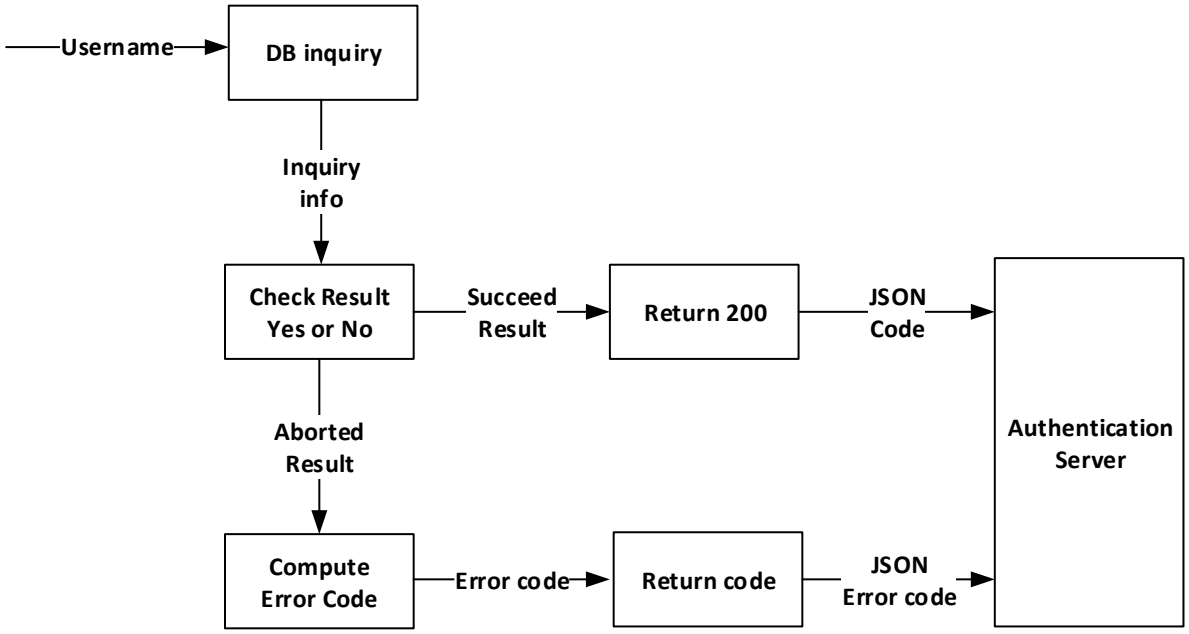

Fig.3. DFD for TEST Module

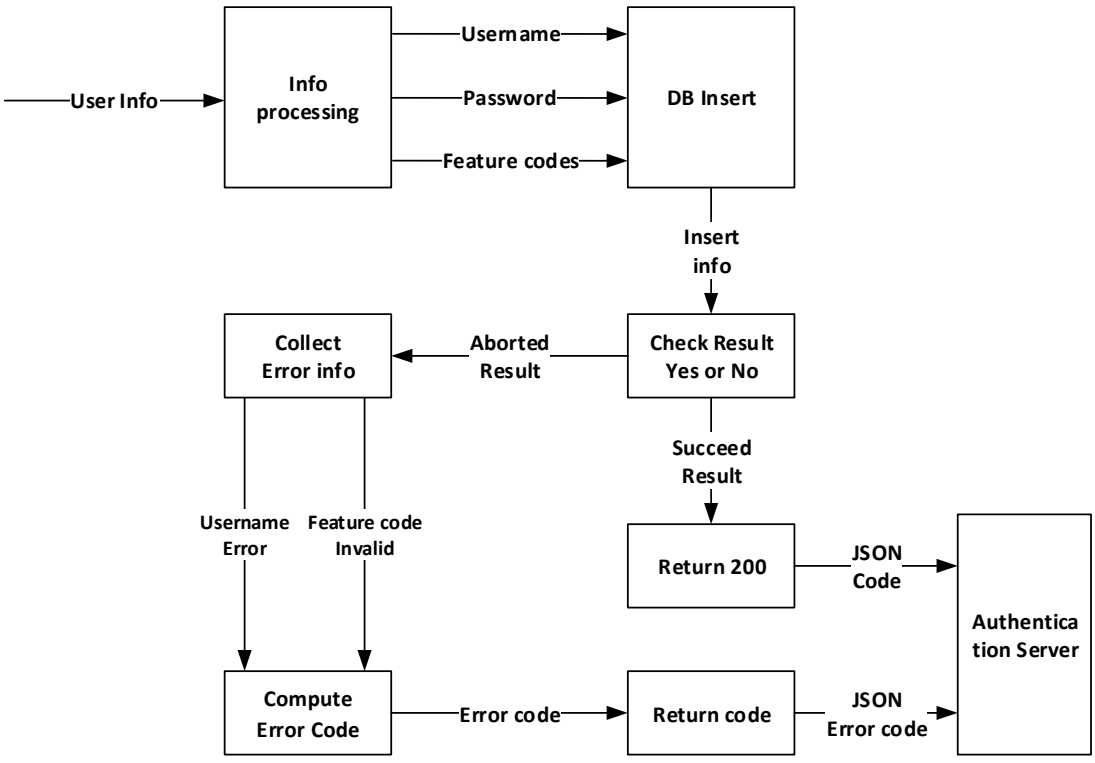

Fig.4. DFD for SIGN UP Module

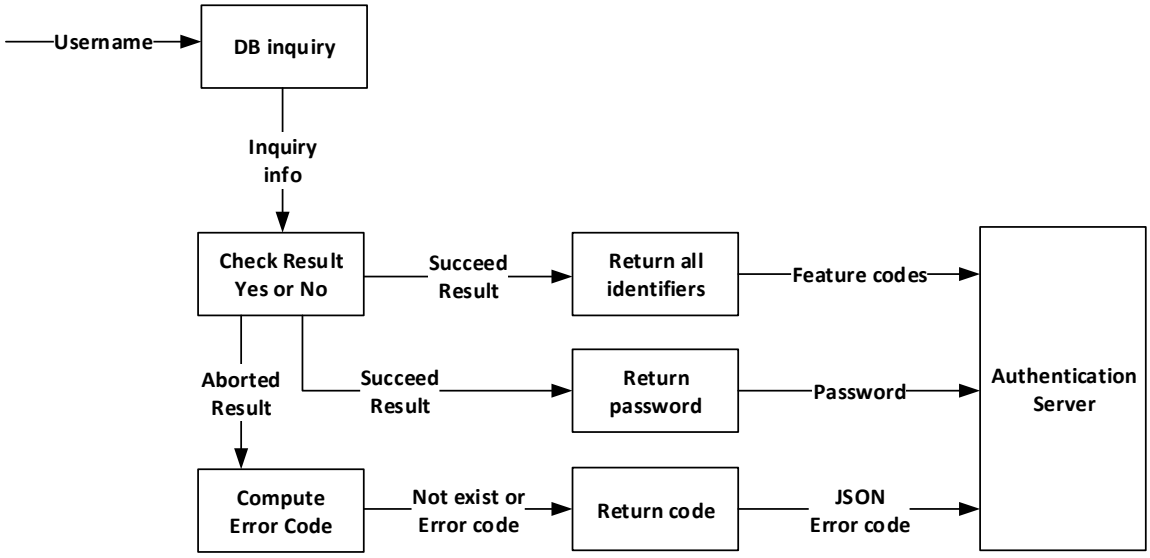

Fig.5. DFD for SIGN IN Module

2.4 Data Dictionary Description

Data Flow Number	1.1
Data Flow Name	Receive Test Request from Server
Description	Receiv Server request which is used to test username validity
Data Flow Composition	Receive Test Request from Server = JSON(userName) userName=[a-zA-Z0-9]{2,20}
Data Flow From	Authentication Server
Data Flow To	Database

Cases:
{
"userName":"helloworld",
}	

Data Flow Number	1.2
Data Flow Name	Send Test Response to Server
Description	Tell server that the username is valid or not, and send test result to serve.
Data Flow Composition	Send Test Response to Server = JSON(retCode) retCode: 200: No errors and user information legal 201: Error, User name already exists 404: Unknown error occurred
Special note	1. If successful, retCode[0] = 200. Otherwise, retCode[0]=-1; 2. Error codes appear only in other locations of the array; 3. All content in retCode is number, not a string
Data Flow From	Database
Data Flow To	Authentication Server

Cases:
{
"retCode":[001,404],
}	

Data Flow Number	1.3
Data Flow Name	Receive Sign Up Request from Server
Description	Receive user registration request from server and create a record for the user in the database.
Data Flow Composition	Receive Sign Up Request from Server = JSON (userName + passwdMd5 + Identifier) userName=[a-zA-Z0-9]{2,20} passwdMd5=[a-zA-Z0-9]{32} Identifier=(webName + id) {1, } webName=[a-zA-Z0-9]{1,} id= [a-zA-Z0-9]{1,32}
Special note	1. Identifier's JSON variable name is *identifiers,* which is a JSON object consisting of a pair (webName + id). (i.e. a group webName and id)
Data Flow From	Authentication Server
Data Flow To	Database

Cases:

```
{
          "userName":"helloworld",
          "passwdMd5": "E10ADC3949BA59ABBE56E057F20F883E",
          "identifiers":
     [
       {
         webName="face++",
         id="aaaaaaaaa"
        },

       {
         webName="gface++",
         id="bbbbbbbb"
        }
     ]
}
```

Data Flow Number	1.4
Data Flow Name	Send Sign Up Response to Server
Description	Return user registration results to server.
Data Flow Composition	Send Sign Up Response to Server = JSON(retCode) retCode: 200: No errors 202: Database insert exception 404: Unknown error occurred
Special note	1. If successful, retCode[0] = 200. Otherwise, retCode[0]=-1; 2. Error codes appear only in other locations of the array; 3. All content in retCode is number, not a string
Data Flow From	Database
Data Flow To	Authentication Server

Cases:

```
{
          "retCode":[001,404]
}
```

Data Flow Number	1.5
Data Flow Name	Receive Sign In Request from Server
Description	Receive login request from server
Data Flow Composition	Receive Sign In Request from Server = JSON (userName) username = [a-zA-Z0-9]{2,20}
Special note	None
Data Flow From	Authentication Server
Data Flow To	Database

Cases:

```
{
        “userName”:”helloworld”
}
```

Data Flow Number	1.6
Data Flow Name	Send Sign In Response to Server
Description	Return user’s login information to server
Data Flow Composition	Send Sign In Response to Server = JSON (passwdMd5 + Identifiers{1, } + retCode) passwdMd5=[a-zA-Z0-9]{32} Identifier=(webName + id) {1, } webName=[a-zA-Z0-9]{1,} id= [a-zA-Z0-9]{1,32} retCode: 200：No errors 203：Error, user name not exists 404：Unknown error occurred
Special note	1. If successful, retCode[0] = 200. Otherwise, retCode[0]=-1; 2. Error codes appear only in other locations of the array; 3. All content in retCode is number, not a string
Data Flow From	Database
Data Flow To	Authentication Server

Cases:

```
{
         “passwdMd5”:” E10ADC3949BA59ABBE56E057F20F883E”,
         "identifiers":
    [
      {
        “webName”=”face++”,
        “id”:=”aaaaaaaaa”
       },

      {
        “webName”=”gface++”,
        “id”=”bbbbbbbb”
       }
    ]
    “retCode”:[200,201,202]
}
```

2.5 Operating Environment

1. Linux : Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-65-generic x86_64)
2. Apache: Apache 2.4.7 (Ubuntu)
3. MySql : mysql Ver 14.14 Distrib 5.5.47
4. PHP : PHP Version 5.5.9-1
5. LAMP : apt-get install lamp-server

3. External Interface Requirements

3.1 Test Module

3.1.1 User interface: Receive username from Authentication server

3.1.2 User interface: Return test result or error code to Authentication server

3.2 Sign up Module

3.2.1 User interface: Receive user info from Authentication server.

3.2.2 User interface: Return Registration result or error code to Authentication server.

3.3 Sign in Module

3.3.1 User interface: Receive username from Authentication server

3.3.2 User interface: Return login info (passwdMd5+identifier) or error code to Server

3.4 Error Code Statement

Error Code	Error Statement
200	No errors
201	Error, User name already exists
202	Database insert exception
203	Error, user name not exists
404	Unknown error occurred